

CLAIMS:

What is claimed is:

- 339
1. A method in a data processing system for executing
5 and processing data in an object oriented environment,
the method comprising the data processing system
implemented steps of:
controlling a presentation of a graphical user
interface using a view controller, wherein the view
10 controller handles user input to the graphical user
interface;
responsive to a selected user input, sending the
selected user input from the view controller to an
application mediator;
15 responsive to receiving the selected user input at
the application mediator, processing the selected user
input at the application mediator;
responsive to a requirement for data, sending a
request from the application mediator to a transporter;
20 responsive to receiving a request from the
application mediator at the transporter, sending the
request to a destination; and
responsive to receiving the request at the
destination, retrieving the data using the destination.
25
 2. The method of claim 1, wherein the graphical user
interface is a component.
 3. The method of claim 1, wherein the graphical user
30 interface is a plurality of components.
 4. The method of claim 1 further comprising:

Accepted for Publication

applying validation rules to user input to validate and format the selected user input.

5 5. The method of claim 1, wherein the destination is a selected destination within the plurality of destinations.

10 6. The method of claim 1, wherein the object oriented environment is Java.

15 7. A method in a data processing system creating an application for processing data in an object oriented system, the method comprising the data processing system implemented steps of:

20 creating a graphical user interface, wherein graphical user interface includes a plurality of components, processes for presenting the plurality of components and receiving user input are handled by a first set of graphical objects, wherein in response to selected user input, a first event is generated;

25 creating an application object, wherein the application process controls a order in which the graphical objects present the set of graphical object containers and process the event and wherein the application generates a second event;

 creating a transport object, wherein the transport object processes the second event and forwards the second event for processing to a destination within the plurality of destinations; and

30 creating a plurality of destination objects, wherein each destination object within the plurality of destinations objects handles accessing a destination

66207730200

within the plurality of destinations.

8. The method of claim 7, wherein the plurality of destinations are a plurality of services located on a server or on the local system to which the data processing system is a client.

9. The method of claim 7, wherein the plurality of destination are a plurality of services located within the data processing system.

10. The method of claim 8, wherein the plurality of services is a plurality of databases.

11. The method of claim 7, wherein the second event is a request for data.

12. The method of claim 7, wherein the second event is a requests to alter data.

13. A data processing system comprising:
a view controller, wherein the view controller handles presentation of a graphical user interface and receives user input;

an application mediator, wherein the application mediator receives selected user input from the view controller and processes the selected user input and generates a request;

a destination, wherein the destination handles the request in response to receiving the request; and

a transporter, wherein the transporter routes the request to the from the application mediator to the

destination.

14. The data processing system of claim 13, wherein the
view controller is a first view controller and further
5 comprising a second view controller, wherein application
mediator controls the first view controller and the
second view controller.

15. A data processing system of claim 13, wherein the
10 destination is a first destination and further
comprising:

a second destination, wherein the second destination
handles the request in response to receiving the request;
and

15 wherein the transporter routes the request to one of
the first destination and the second destination based on
contents within the request.

16. A data processing system for executing and
20 processing data in an object oriented environment, the
data processing system comprising:

controlling means for controlling a presentation of
a graphical user interface using a view controller,
wherein the view controller handles user input to the
25 graphical user interface;

sending means for responsive to a selected user
input, for sending the selected user input from the view
controller to an application mediator;

processing means, responsive to receiving the
30 selected user input at the application mediator, for
processing the selected user input at the application
mediator;

SECRET

sending means, responsive to a requirement for data, for sending a request from the application mediator to a transporter;

5 sending means, responsive to receiving a request from the application mediator at the transporter, for sending the request to a destination; and

receiving means, responsive to receiving the request at the destination, for retrieving the data using the destination.

10

17. The data processing system of claim 16, wherein the graphical user interface is a component.

15

18. The data processing system of claim 16, wherein the graphical user interface is a plurality of components.

19. The data processing system of claim 16 further comprising:

20

applying means for applying validation rules to user input to validate and format the selected user input.

20. The data processing system of claim 16, wherein the destination is a selected destination within the plurality of destinations.

25

21. The data processing system of claim 16, wherein the object oriented environment is Java.

30

22. A data processing system creating an application for processing data in an object oriented system, the data processing system comprising:

first creating means for creating a graphical user

interface, wherein graphical user interface includes a plurality of components, processes for presenting the plurality of components and receiving user input are handled by a first set of graphical objects, wherein in
5 response to selected user input, a first event is generated;

second creating means for creating an application object, wherein the application process controls a order in which the graphical objects present the set of
10 graphical object containers and process the event and wherein the application generates a second event;

third creating means for creating a transport object, wherein the transport object processes the second event and forwards the second event for processing to a
15 destination within the plurality of destinations; and

fourth creating means for creating a plurality of destination objects, wherein each destination object within the plurality of destinations objects handles accessing a destination within the plurality of
20 destinations.

23. The data processing system of claim 22, wherein the plurality of destinations are a plurality of services located on a server or on the local system to which the
25 data processing system is a client.

24. The data processing system of claim 22, wherein the plurality of destination are a plurality of services located within the data processing system.
30

25. The data processing system of claim 24, wherein the plurality of services is a plurality of databases.

26. The data processing system of claim 22, wherein the second event is a request for data.

5 27. The data processing system of claim 22, wherein the second event is a requests to alter data.

28. A computer program product in a computer readable medium for executing and processing data in an object oriented environment, the method comprising the computer program product implemented steps of:

10 first instructions for controlling a presentation of a graphical user interface using a view controller, wherein the view controller handles user input to the graphical user interface;

15 second instructions, responsive to a selected user input, for sending the selected user input from the view controller to an application mediator;

20 third instructions, responsive to receiving the selected user input at the application mediator, for processing the selected user input at the application mediator;

25 fourth instructions, responsive to a requirement for data, for sending a request from the application mediator to a transporter;

fifth instructions, responsive to receiving a request from the application mediator at the transporter, for sending the request to a destination; and

30 sixth instructions, responsive to receiving the request at the destination, for retrieving the data using the destination.

SECRET - 11/3/2000

29. A computer program product in a computer readable medium for creating an application for processing data in an object oriented system, the method comprising the computer program product implemented steps of:

5 first instructions for creating a graphical user interface, wherein graphical user interface includes a plurality of components, processes for presenting the plurality of components and receiving user input are handled by a first set of graphical objects, wherein in
10 response to selected user input, a first event is generated;

second instructions for creating an application object, wherein the application process controls a order in which the graphical objects present the set of
15 graphical object containers and process the event and wherein the application generates a second event;

third instructions for creating a transport object, wherein the transport object processes the second event and forwards the second event for processing to a
20 destination within the plurality of destinations; and

fourth instructions for creating a plurality of destination objects, wherein each destination object within the plurality of destinations objects handles accessing a destination within the plurality of
25 destinations.

388 30. A method in a data processing system for refreshing data in an application, the method comprising the data processing system implemented steps of:

30 receiving a change in data at an application mediator, wherein the application mediator handles a plurality of view controllers and a plurality of

application mediators;

sending a call to each application mediator within the plurality of application mediators to refresh data in objects associated with the plurality of application

5 mediators;

sending a call to each view controller within the plurality of view controllers to refresh data in objects associated with the plurality of view controllers;

responsive to receiving a call at an application
10 mediator within the plurality of application mediators, updating the data in an object associated with the application mediator, wherein a call is made to each view controller to refresh the data; and

responsive to receiving a call at a view controller
15 within the plurality of view controllers, updating the data in an object associated with the view controller, wherein a display of containers presented by the view controller is updated.

20 31. The method of claim 30 further comprising:

determining, prior to updating the data, whether the data is a type recognized and handled by the view controller;

responsive to a determination that the data is an
25 unrecognized type and handled, formatting the data into at least one recognized types by the view controller; and

responsive to a determination that the data is an unrecognized type and unhandled, ignoring the data or
30 yielding an error.

32. The method of claim 30, wherein the data causes a

change in a configuration of components displayed in the container.

33. The method of claim 30, wherein the data contains a
5 change in permissions resulting in a change in components
accessible through a user input.

34. The method of claim 30, wherein each of the
10 plurality of view controllers handles presentation of a
container.

35. The method of claim 30, wherein only a single view
controller presents a container at a time.

36. The method of claim 30, wherein the container is a
15 component.

37. The method of claim 30, wherein the container is a
20 bean.

38. The method of claim 30, wherein the data is an
object.

39. The method of claim 30, wherein the data is an
25 object containing multiple objects.

40. A method in a data processing system for refreshing
data in an application, the method comprising the data
processing system implemented steps of:

30 receiving a call to update data in the application,
wherein the data is destined for a component in the
application;

identifying a data type for the data; and
responsive to the data type being a handled data
type, formatting the data and calling a refresh on the
component.

5

41. The method of claim 40, wherein handled data type is
a singleton.

10

42. The method of claim 40, wherein the handled data
type is an aggregate.

15

43. The method of claim 40 further comprising:
responsive to the data type being an unhandled data
type, determining whether the data contains multiple
object;

20

responsive to the data containing a set of objects,
formatting each object within the set of objects into a
recognized format; and
calling a refresh on the component.

25

44. The method of claim 40, wherein the component is a
view controller.

45. The method of claim 40, wherein the component is an
application mediator

30

46. A data processing system for refreshing data in an
application, the data processing system comprising:
receiving means for receiving a change in data at an
application mediator, wherein the application mediator
handles a plurality of view controllers and a plurality
of application mediators;

first sending means for sending a call to each application mediator within the plurality of application mediators to refresh data in objects associated with the plurality of application mediators;

- 5 second sending means for sending a call to each view controller within the plurality of view controllers to refresh data in objects associated with the plurality of view controllers;

- 10 first updating means, responsive to receiving a call at an application mediator within the plurality of application mediators, for updating the data in an object associated with the application mediator, wherein a call is made to each view controller to refresh the data; and

- 15 second updating means, responsive to receiving a call at a view controller within the plurality of view controllers, for updating the data in an object associated with the view controller, wherein a display of containers presented by the view controller is updated.

- 20 47. The data processing system of claim 46 further comprising:

determining means for determining, prior to updating the data, whether the data is a type recognized and handled by the view controller;

- 25 formatting means, responsive to a determination that the data is an unrecognized type and handled, for formatting the data into at least one recognized types by the view controller; and

- 30 ignoring means, responsive to a determination that the data is an unrecognized type and unhandled, for ignoring the data or yielding an error.

48. The data processing system of claim 46, wherein the data causes a change in a configuration of components displayed in the container.

5 49. The data processing system of claim 46, wherein the data contains a change in permissions resulting in a change in components accessible through a user input.

10 50. The data processing system of claim 46, wherein each of the plurality of view controllers handles presentation of a container.

15 51. The data processing system of claim 46, wherein only a single view controller presents a container at a time.

52. The data processing system of claim 46, wherein the container is a component.

20 53. The data processing system of claim 46, wherein the container is a bean.

54. The data processing system of claim 46, wherein the data is an object.

25 55. The data processing system of claim 46, wherein the data is an object containing multiple objects.

30 56. A data processing system for refreshing data in an application, the data processing system comprising:
receiving means for receiving a call to update data in the application, wherein the data is destined for a component in the application;

identifying means for identifying a data type for the data; and

formatting means, responsive to the data type being a handled data type, for formatting the data and calling
5 a refresh on the component.

57. The data processing system of claim 56, wherein handled data type is a singleton.

10 58. The data processing system of claim 56, wherein the handled data type is an aggregate.

59. The data processing system of claim 56 further comprising:

15 determining means, responsive to the data type being an unhandled data type, for determining whether the data contains multiple object;

formatting means, responsive to the data containing a set of objects, for formatting each object within the
20 set of objects into a recognized format; and

calling means for calling a refresh on the component.

60. The data processing system of claim 56, wherein the
25 component is a view controller.

61. The data processing system of claim 56, wherein the component is an application mediator

30 62. A computer program product in a computer readable medium for refreshing data in an application, the computer program product comprising:

first instructions for receiving a change in data at an application mediator, wherein the application mediator handles a plurality of view controllers and a plurality of application mediators;

5 second instructions for sending a call to each application mediator within the plurality of application mediators to refresh data in objects associated with the plurality of application mediators;

10 third instructions for sending a call to each view controller within the plurality of view controllers to refresh data in objects associated with the plurality of view controllers;

15 fourth instructions, responsive to receiving a call at an application mediator within the plurality of application mediators, for updating the data in an object associated with the application mediator, wherein a call to each view controller to refresh the data is also made; and

20 fifth instructions, responsive to receiving a call at a view controller within the plurality of view controllers, for updating the data in an object associated with the view controller, wherein a display of containers presented by the view controller is updated.

25 63. A computer program product in a computer readable medium for refreshing data in an application, the computer program product comprising:

30 first instructions for receiving a call to update data in the application, wherein the data is destined for a component in the application;

second instructions for identifying a data type for the data; and

2025 RELEASE UNDER E.O. 14176

389
5

10

controlling a location of the component within the display using a second object, wherein the second object controls the location of the component in response to receiving an event; and

15

65. The method of claim 64, wherein the first object is a view controller, the second object is a placement listener, and the third object is an application mediator.

20

25

controlling a location of each of the plurality of containers using a placement object, wherein the placement object places the containers in the graphical user interface in response to receiving events; and

30

generating events using an application mediator object, wherein the events are sent to the placement object.

1. The first step is to identify the problem or question that needs to be addressed. This involves understanding the context and the specific requirements of the task.

67. The method of claim 66, wherein the container is a panel.
- 5 68. The method of claim 66, wherein the container is a button.
69. A display mechanism for use in a data processing system to display a container in a display in the data
10 processing system, the display mechanism comprising:
a first object used to display a graphical user interface in the display and to receive user input;
a second object used to position the graphical user interface in the display in response to receiving an
15 event; and
a third object used to generate the events.
70. The display mechanism of claim 69, wherein the first
20 object is a display object and the second object is a positioning object.
71. The display mechanism of claim 70, wherein the display object is an instance of a view controller.
- 25 72. The display mechanism of claim 70, wherein the positioning object is an instance of a placement listener.
73. The display mechanism of claim 69, wherein the
30 display mechanism is implemented in Java.
74. The display mechanism of claim 69, wherein the

2025 RELEASE UNDER E.O. 14176

Docket No. 9-99-339

second object is useable with a plurality of the first objects.

75. The display mechanism of claim 69, wherein the first
5 object is a view controller, the second object is a placement listener, and the third object is an application mediator.

76. The display mechanism of claim 69, wherein the
10 application mediator generates an event in response to a user input to the container.

77. A data processing system for displaying a component or container, the system comprising:

15 displaying means for displaying the component within a display using a first object;

controlling means for controlling a location of the component within the display using a second object, wherein the second object controls the location of the
20 component in response to receiving an event; and

displaying means for selectively displaying the component using a third object, wherein the third object generates the event.

25 78. The system of claim 77, wherein the first object is a view controller, the second object is a placement listener, and the third object is an application mediator.

30 79. A data processing system for displaying a graphical user interface, the system comprising:

displaying means for displaying a container for a

66207-1130460

5

10

15

20

25

390

30

83. A process in a data processing system for providing access to a set of host services, the method comprising the data processing system implemented steps of:

5 responsive to a selected user input, sending the
selected user input from the view controller to an
application mediator;

responsive to receiving the selected user input at
the application mediator, processing the selected user
10 input at the application mediator;

responsive to the application mediator determining that a service from a set of host services is required, generating an event; and

responsive to detecting the event at a listener
15 object, executing a method in the listener object to
perform the service, wherein interaction with the set of
host services is provided from the graphical user
interface.

20 84. The process of claim 83, wherein the set of host
services is a desktop program.

85. The process of claim 83, wherein the service
executes within an operating system located on the data
25 processing system, wherein the service shuts down the
application.

86. The process of claim 83, wherein the service launches an application.

30

87. The process of claim 83, wherein the service sends a string message for display in the desktop environment.

88. The method of claim 83, wherein the service sends a title message for display in a window in the desktop environment.

5

89. The process of claim 83, wherein the event includes a major code identifying a type of event.

10

90. The process of claim 89, wherein the event includes a minor code identifying additional information about the type of event.

15

91. The process of claim 89, wherein the event includes an identification of the source of the event.

92. The process of claim 83, wherein the event includes data to be passed to the service.

20

93. A process in a data processing system for managing services in a desktop environment from an object oriented-environment, the process comprising the data processing system implemented steps of:

25

controlling a presentation of a graphical user interface using a view controller, wherein the view controller handles user input to the graphical user interface;

responsive to a selected user input, sending the selected user input from the view controller to an application mediator;

30

responsive to receiving the selected user input at the application mediator, processing the selected user input at the application mediator;

responsive to detecting the event at a listener
5 object, executing a method in the listener object to
perform the service in the desktop environment.

10

15

20

25

30

SECRET

102. A data processing system comprising:

a desktop environment;

an object oriented environment, wherein the object oriented environment includes:

5 a plurality of view controllers, wherein the plurality of view controllers control display of a plurality of containers, such that each view controller controls display of a single container and handles user input to that container;

10 an application mediator, wherein the application mediator controls the view controllers to control display of the view controllers, processes data, and generates an event requiring a service from the desktop environment; and

15 a toplistener, wherein the detects the event and executes a process to invoke the service.

103. A data processing system for providing access to a set of host services, the data processing system comprising:

controlling means for controlling a presentation of a graphical user interface using a view controller, wherein the view controller handles user input to the graphical user interface;

25 sending means, responsive to a selected user input, for sending the selected user input from the view controller to an application mediator;

receiving means, responsive to receiving the selected user input at the application mediator, for processing the selected user input at the application mediator;

30 generating means, responsive to the application

SECRET

mediator determining that a service from a set of host services is required, for generating an event; and
executing means, responsive to detecting the event at a listener object, for executing a method in the
5 listener object to perform the service, wherein interaction with the set of host services is provided from the graphical user interface.

104. The data processing system of claim 103, wherein the
10 set of host services is a desktop program.

105. The data processing system of claim 103, wherein the service executes within an operating system located on the data processing system, wherein the service shuts
15 down the application.

106. The data processing system of claim 103, wherein the service launches an application.

20 107. The data processing system of claim 103, wherein the service sends a string message for display in the desktop environment.

108. The data processing system of claim 103, wherein
25 the service sends a title message for display in a window in the desktop environment.

109. The data processing system of claim 103, wherein the event includes a major code identifying a type of event.
30

110. The data processing system of claim 109, wherein the event includes a minor code identifying additional

SECRET

information about the type of event.

111. The data processing system of claim 109, wherein the event includes an identification of the source of the event.

112. The data processing system of claim 103, wherein the event includes data to be passed to the service.

113. A data processing system for managing services in a desktop environment from an object oriented-environment, the data processing system comprising:

controlling means for controlling a presentation of a graphical user interface using a view controller, wherein the view controller handles user input to the graphical user interface;

sending means, responsive to a selected user input, for sending the selected user input from the view controller to an application mediator;

processing means, responsive to receiving the selected user input at the application mediator, for processing the selected user input at the application mediator;

generating means, responsive to the application mediator determining that a service is required in the desktop environment, for generating an event; and

executing means, responsive to detecting the event at a listener object, for executing a method in the listener object to perform the service in the desktop environment.

114. The data processing system of claim 113, wherein the

Docket No. 9-99-339

process shuts down an application in the desktop environment.

5 115. The data processing system of claim 113, wherein the data processing system launches an application in the desktop environment.

10 116. The data processing system of claim 113, wherein the data processing system sends a string message for display in the desktop environment.

15 117. The data processing system of claim 113, wherein the data processing system sends title message for display in a window.

118. The data processing system of claim 113, wherein the event includes a major code identifying a type of event.

20 119. The data processing system of claim 118, wherein the event includes a minor code identifying additional information about the type of event.

25 120. The data processing system of claim 118, wherein the event includes an identification of the source of the event.

30 121. The data processing system of claim 113, wherein the event includes data to be passed to the desktop environment.

122. A computer program product in a computer readable medium for use in a data processing system for providing

DOCKET "H" 9-99-339

access to a set of host services, the computer program product comprising:

first instructions controlling a presentation of a graphical user interface using a view controller, wherein
5 the view controller handles user input to the graphical user interface;

second instructions, responsive to a selected user input, for sending the selected user input from the view controller to an application mediator;

10 third instructions, responsive to receiving the selected user input at the application mediator, for processing the selected user input at the application mediator;

fourth instructions, responsive to the application
15 mediator determining that a service from a set of host services is required, for generating an event; and

fifth instructions, responsive to detecting the event at a listener object, for executing a method in the listener object to perform the service, wherein
20 interaction with the set of host services is provided from the graphical user interface.

123. A computer program product in a computer readable medium for use in a data processing system for managing
25 services in a desktop environment from an object oriented-environment, the computer program product comprising:

first instructions for controlling a presentation of a graphical user interface using a view controller,
30 wherein the view controller handles user input to the graphical user interface;

second instructions, responsive to a selected user

2025 RELEASE UNDER E.O. 14176

input, for sending the selected user input from the view controller to an application mediator;

third instructions, responsive to receiving the selected user input at the application mediator, for
5 processing the selected user input at the application mediator;

fourth instructions, responsive to the application mediator determining that a service is required in the desktop environment, for generating an event; and

10 fifth instructions, responsive to detecting the event at a listener object, for executing a method in the listener object to perform the service in the desktop environment.

15 124. A method in a data processing system for managing requests, the method comprising the data processing system implemented steps of:

receiving a request event at a transporter object, wherein the request event is self identifying through its
20 type, a major code, a minor code, and object data.

identifying a destination object within the plurality of destination objects using the request event to form an identified destination object; and

25 sending the request event to the identified destination object, wherein the identified destination object handles the request using the indication and accesses the target.

125. The method of claim 124, wherein the target is a
30 service.

126. The method of claim 125, wherein the service is

662204-1633468

391

DC

Docket No. 99-99-339

located on a remote data processing system.

127. The method of claim 124, wherein the transporter receives the request event from an application mediator.

5

128. The method of claim 124, wherein the indication is to access a service at a remote location and further comprising:

responsive to receiving the request at the destination object, accessing the service at the remote location using the destination object, wherein the destination object formats the request into one recognizable by the remote server to access the service.

129. The method of claim 128 further comprising:
receiving a response from the remote service ;
formatting the response into a request event; and
returning the request event to the transporter.

130. The method of claim 129, wherein the request event includes the data.

131. The method of claim 129, wherein the remote service is a database.

25

132. A data processing system comprising:

a plurality of destination objects, wherein responsive to receiving a request event having a first indication and a second indication, a destination object within the plurality of destination objects identifies a function to perform on request event, wherein the function is identified from a second indication in the

66001-1130000

request event; and

a transporter object, wherein, responsive to receiving the request event, the transporter object identifies the destination within the plurality of destinations from the first indication, and routes the request to the destination.

133. The data processing system of claim 132, wherein the first indication is a major code and the second indication is a minor code.

134. The data processing system of claim 133, wherein the major code is a class name of the destination object and the minor code is a method name that is to be invoked.

135. The data processing system of claim 132, wherein the request event includes data.

136. A data processing system for managing requests, the data processing system comprising:

receiving means for receiving a request event at a transporter object, wherein the request event is self identifying through its type, a major code, a minor code, and object data.

identifying means for identifying a destination object within the plurality of destination objects using the request event to form an identified destination object; and

sending means for sending the request event to the identified destination object, wherein the identified destination object handles the request using the indication and accesses the target.

66207-11800000

VN
9/27/03
137
136. The data processing system of claim 136, wherein the target is a service.

5 138. The data processing system of claim 137, wherein the service is located on a remote data processing system.

139. The data processing system of claim 136, wherein the transporter receives the request event from an
10 application mediator.

140. The data processing system of claim 136, wherein the indication is to access a service at a remote location and further comprising:

15 accessing means, responsive to receiving the request at the destination object, for accessing the service at the remote location using the destination object, wherein the destination object formats the request into one recognizable by the remote server to access the service.

20 141. The data processing system of claim 140 further comprising:

 second receiving means for receiving a response from the remote service;

25 formatting means for formatting the response into a request event; and

 returning means for returning the request event to the transporter.

30 142. The data processing system of claim 141, wherein the request event includes the data.

66201-11306100

Docket No. 99-339

143. The data processing system of claim 141, wherein the remote service is a database.

144. A computer program product in a computer readable medium for use in data processing system for managing requests, the computer program product comprising:

first instructions for receiving a request event at a transporter object, wherein the request event is self identifying through its type, a major code, a minor code, and object data.

second instructions for identifying a destination object within the plurality of destination objects using the request event to form an identified destination object; and

third instructions for sending the request event to the identified destination object, wherein the identified destination object handles the request using the indication and accesses the target.

392
20 145. A method in a data processing system for displaying a graphical user interface, the method comprising the data processing system implemented steps of:

displaying a container in a graphical user interface from a set of containers, wherein a display of the container handled by a view controller from a set of view controllers, wherein each view controller handles the display of an associated container within the set of containers and user input for the associated container; and

30 altering a display of the set of containers by an application mediator, wherein the set of containers are displayed in an order determined by the application

660007-1130460

mediator.

146. The method of claim 145 further comprising:
generating an event by a view controller in response
5 to a selected user input to a container handled by the
view controller, wherein the altering step is responsive
to the application mediator receiving the event.

147. The method of claim 145, wherein the set of
10 containers is a set of panels.

148. A method in a data processing system for displaying
a set of containers, the method comprising the data
processing system implemented steps of:
15 displaying a current container from a set of
containers using a plurality of view controllers, wherein
each view controller handles user input for a container
within the set of containers;
responsive to a selected user input to the current
20 container, sending the selected user input from the view
controller to an application mediator; and
responsive to receiving the selected user input at
the application mediator, sending a response to the
plurality of view controllers to display a different
25 container within the set of containers.

149. The method of claim 148, wherein the set of
containers is a set of panels.

30 150. The method of claim 148, wherein the selected input
is a first selected user input and further comprising:
responsive to a second selected user input to the

current container, sending the second selected user input from the view controller to an application mediator; and responsive to receiving the second selected user input at the application mediator, sending a view controller for the current container to alter a presentation of the current container.

151. A data processing system comprising:
a plurality of containers, wherein a current container is displayed;
a plurality of view controllers, wherein the view controllers control a display of the plurality of containers in which each view controller controls a single container within the plurality of containers and generates an event in response to a selected user input; and
an application mediator, wherein the application mediator handles the selected event and sends a response to a view controller within the plurality of view controllers, causing another container to be displayed in place of the current container.

152. The data processing system of claim 151, wherein the application mediates access to a service.

153. The data processing system of claim 152, wherein the service is a database.

154. A data processing system for displaying a graphical user interface, the data processing system comprising:
displaying means for displaying a container in a graphical user interface from a set of containers,

wherein a display of the container handled by a view controller from a set of view controllers, wherein each view controller handles the display of an associated container within the set of containers and user input for the associated container; and

altering means for altering a display of the set of containers by an application mediator, wherein the set of containers are displayed in an order determined by the application mediator.

155. The data processing system of claim 154 further comprising:

generating means for generating an event by a view controller in response to a selected user input to a container handled by the view controller, wherein the altering step is responsive to the application mediator receiving the event.

156. The data processing system of claim 154, wherein the set of containers is a set of panels.

157. A data processing system for displaying a set of containers, the data processing system comprising:

displaying means for displaying a current container from a set of containers using a plurality of view controllers, wherein each view controller handles user input for a container within the set of containers;

sending means, responsive to a selected user input to the current container, for sending the selected user input from the view controller to an application mediator; and

sending means, responsive to receiving the selected

2008-11-13 14:00:00

5 158. The data processing system of claim 157, wherein the set of containers is a set of panels.

10 comprising:

15 sending means, responsive to receiving the second selected user input at the application mediator, for sending a view controller for the current container to alter a presentation of the current container.

first instructions for displaying a container in a graphical user interface from a set of containers, wherein a display of the container handled by a view controller from a set of view controllers, wherein each view controller handles the display of an associated container within the set of containers and user input for the associated container; and

second instructions for altering a display of the set of containers by an application mediator, wherein the

[illegible]

set of containers are displayed in an order determined by the application mediator.

161. A computer program product in a computer readable
5 medium for displaying a set of containers, the method comprising the computer program product implemented steps of:

first instructions for displaying a current
container from a set of containers using a plurality of
10 view controllers, wherein each view controller handles user input for a container within the set of containers;

second instructions, responsive to a selected user
input to the current container, for sending the selected
user input from the view controller to an application
15 mediator; and

third instructions, responsive to receiving the
selected user input at the application mediator, for
sending a response to the plurality of view controllers
to display a different container within the set of
20 containers.

393
162. A method in a data processing system for performing
validation of user input, the method comprising the data
processing system implemented steps of:

25 receiving user input to a container displayed in a graphical user interface, wherein display of the container and the user input to the container are handled by a view controller;

responsive to receiving the user input, sending a
30 call to a validation object; and

responsive to receiving the call, testing, by the validation object, the user input using a set of

Accepted for Filing

validation rules.

163. The method of claim 162, wherein a validation rule
in the set of validation rule is associated with another
5 validation rule in the set of chained validation rules
and passes the data to the another validation rule in
response to the user input being validated the validation
rule

10 164. The method of claim 162, wherein the step of testing
comprises:

placing data from the user input to be validated in
a data structure; and

15 iteratively testing the data using the set of
validation rules.

165. The method of claim 162, wherein the set of
validation rules is a set of associated validation rules.

20 166. The method of claim 162, wherein the set of
associated validation rules are associated by chaining
each validation rule within the set of associated
validation rules with another validation rule within the
set of validation rules.

25 167. A method in a data processing system for performing
validation of user input, the method comprising the data
processing system implemented steps of:

30 receiving user input in a container displayed in a
graphical user interface, wherein presentation of the
container and the user input to the container are handled
by a view controller;

168. The method of claim 167, wherein the criteria is a rule.

10. 169. The method of claim 167, wherein the criteria is a set of rules.

170. The method of claim 167, wherein the set of rules are a chained set of rules.

15

171. The method of claim 167, wherein the call is a first type of call and further comprising:

responsive to a second type of call, formatting, by the validation object, the user input.

20 172. The method of claim 167, wherein the container is a component.

173. A data processing system comprising:

25 a view controller, wherein the view controller handles a display of a container in a graphical user interface and handles user input to the container; and

 a validation object, wherein the validation object includes a rule used to test data in response to a call

30 from the view controller.

174. A data processing system for performing validation

[illegible]

of user input, the data processing system comprising:

receiving means for receiving user input to a container displayed in a graphical user interface, wherein display of the container and the user input to

5 the container are handled by a view controller;

sending means, responsive to receiving the user input, for sending a call to a validation object; and

10 testing means, responsive to receiving the call, for testing, by the validation object, the user input using a set of validation rules.

175. The data processing system of claim 174, wherein a validation rule in the set of validation rule is associated with another validation rule in the set of
15 chained validation rules and passes the data to the another validation rule in response to the user input being validated the validation rule.

176. The data processing system of claim 174, wherein the
20 testing means comprises:

placing means for placing data from the user input to be validated in a data structure; and

25 testing means for iteratively testing the data using the set of validation rules.

177. The data processing system of claim 174, wherein the set of validation rules is a set of associated validation rules.

30 178. The data processing system of claim 174, wherein the set of associated validation rules are associated by chaining each validation rule within the set of

2025 RELEASE UNDER E.O. 14176

associated validation rules with another validation rule within the set of validation rules.

179. A data processing system for performing validation of
5 user input, the data processing system comprising:

receiving means for receiving user input in a container displayed in a graphical user interface, wherein presentation of the container and the user input to the container are handled by a view controller;

10 sending means, responsive to receiving the user input, for sending, by the view controller, a call to a validation object; and

testing means, responsive to the call, for testing, by the validation object, the user input using a
15 criteria, wherein the rule is separate from the view controller.

180. The data processing system of claim 179, wherein the criteria is a rule.

20

181. The data processing system of claim 179, wherein the criteria is a set of rules.

182. The data processing system of claim 179, wherein the
25 set of rules are a chained set of rules.

183. The data processing system of claim 179, wherein the call is a first type of call and further comprising:

formatting means, responsive to a second type of
30 call, formatting, by the validation object, the user input.

66004-1130446

184. The data processing system of claim 167, wherein the container is a component.

185. A computer program product in a computer readable
5 medium for use in a data processing system for performing validation of user input, the computer program product comprising:

first instructions for receiving user input to a container displayed in a graphical user interface,
10 wherein display of the container and the user input to the container are handled by a view controller;

second instructions, responsive to receiving the user input, for sending a call to a validation object; and

15 third instructions, responsive to receiving the call, for testing, by the validation object, the user input using a set of validation rules.

186. A computer program product in a computer readable
20 medium for use in a data processing system for performing validation of user input, the computer program product comprising:

first instructions for receiving user input in a container displayed in a graphical user interface, wherein
25 presentation of the container and the user input to the container are handled by a view controller;

second instructions, responsive to receiving the user input, for sending, by the view controller, a call to a validation object; and

30 third instructions, responsive to the call, for testing, by the validation object, the user input using a criteria, wherein the rule is separate from the view

2025 RELEASE UNDER E.O. 14176

394
controller.

187. A method in a data processing system for managing permissions in an application, the method comprising the
5 data processing system implemented steps of:

receiving a user input changing a security level for the application at a container handled by a view controller;

generating a view event describing the user input;
10 receiving the view event at an application mediator;
responsive to receiving the view event, sending a request event in response to the receiving the view event; and

receiving at the application mediator, a permission
15 corresponding to the security level, wherein the permission alters an item in the application.

188. The method of claim 187, wherein the item is content in the container and further comprising:

20 sending the permission to the view controller, wherein the permission selectively alters content within the container.

189. The method of claim 187, wherein the item is a
25 function in the application mediator.

190. The method of claim 187, wherein the user input changing the security level is a user login to the application.

30 191. The method of claim 187, wherein the content altered using the set of permissions is an enablement of a

66601-1180260

function.

192. The method of claim 187, wherein the content altered using the set of permissions is a disablement of a
5 function.

193. The method of claim 187, wherein the set of permissions is a set of key/value pairs, wherein a key is used to identify content and a value is used to identify
10 a setting for the content.

194. A method in a data processing system for managing permissions in an application, the method comprising the data processing system implemented steps of:
15 receiving a user input at a container handled by a view controller, wherein the user input requests a change in permissions in the application;
generating a view event describing the user input;
receiving the view event at an application mediator;
20 responsive to receiving the view event, retrieving by the application mediator, a set of permissions corresponding to the user input;
sending the set of permissions to the view controller; and
25 selectively altering content within the container using the set of permissions.

195. The method of claim 194, wherein the step of selectively altering the content includes:
30 enabling content within the container.

196. The method of claim 195, wherein the content enabled

Docket No. 9-99-339

is a ~~button~~.

197. The method of claim 194, wherein the step of selectively altering the contents includes:

5 disabling content within the container.

198. The method of claim 196, wherein the content disabled is a text field.

10 199. The method of claim 194, wherein the set of permissions includes a plurality of key and value pairs, wherein a key in a key and value pair identifies an alterable item and a value in the key and value pair identifies a setting for the alterable item.

15 200. The method of claim 194, wherein the user input is a selection of a button.

201. The method of claim 194, wherein the set of
20 permissions are associated with a user profile.

202. The method of claim 194, wherein the user input is a change in security.

25 203. A data processing system comprising:

a view controller, wherein the view controller handles presenting a container, handles input to the container, generates a view event in response to a selected user input, and alters content in the container in response to a change in permissions; and

30 in response to a change in permissions; and
an application mediator, wherein the application
mediator receives the view event, obtains permissions for

content in the container in response to the view event indicating a request to change permissions for the content, and sends the permissions to the view controller to set the permissions in the view controller.

5

204. The data processing system of claim 203, wherein the container is a panel.

10

205. The data processing system of claim 203, wherein the permissions are a set of key/value pairs, wherein a key in a key/value pair identifies alterable content and a value in the key/value pair identifies a setting for the alterable content.

15

206. The data processing system of claim 203, wherein permissions are associated with a user profile.

207. The data processing system of claim 203 further comprising:

20

a plurality of view controllers, wherein each view controller within the plurality of view controllers handles a container and wherein the container includes content, which is alterable by permissions, and wherein the application mediator handles permissions for the plurality of view controllers.

25

208. A data processing system for managing permissions in an application, the data processing system comprising:

30

first receiving means for receiving a user input changing a security level for the application at a container handled by a view controller;
generating means for generating a view event

describing the user input;

second receiving means for receiving the view event at an application mediator;

5 sending means, responsive to receiving the view event, for sending a request event in response to the receiving the view event; and

10 third receiving means for receiving at the application mediator, a permission corresponding to the security level, wherein the permission alters an item in the application.

209. The data processing system of claim 208, wherein the item is content in the container and further comprising:

15 sending means for sending the permission to the view controller, wherein the permission selectively alters content within the container.

210. The data processing system of claim 208, wherein the item is a function in the application mediator.

211. The data processing system of claim 208, wherein the user input changing the security level is a user login to the application.

25 212. The data processing system of claim 208, wherein the content altered using the set of permissions is an enablement of a function.

30 213. The data processing system of claim 208, wherein the content altered using the set of permissions is a disablement of a function.

2025 RELEASE UNDER E.O. 14176

214. The data processing system of claim 208, wherein the set of permissions is a set of key/value pairs, wherein a key is used to identify content and a value is used to identify a setting for the content.

5

215. A data processing system for managing permissions in an application, the data processing system comprising:

first receiving means for receiving a user input at a container handled by a view controller, wherein the user input requests a change in permissions in the application;

generating means for generating a view event describing the user input;

second receiving means for receiving the view event at an application mediator;

retrieving means, responsive to receiving the view event, for retrieving by the application mediator, a set of permissions corresponding to the user input;

sending means for sending the set of permissions to the view controller; and

altering means for selectively altering content within the container using the set of permissions.

216. The data processing system of claim 215, wherein the means of selectively altering the content includes:

enabling means for enabling content within the container.

217. The data processing system of claim 216, wherein the content enabled is a button.

218. The data processing system of claim 215, wherein the

Accepted for Settlement

means of selectively altering the contents includes:
disabling means for disabling content within the
container.

5 219. The data processing system of claim 217, wherein the
content disabled is a text field.

220. The data processing system of claim 215, wherein the
set of permissions includes a plurality of key and value
10 pairs, wherein a key in a key and value pair identifies
an alterable item and a value in the key and value pair
identifies a setting for the alterable item.

221. The data processing system of claim 215, wherein the
15 user input is a selection of a button.

222. The data processing system of claim 215, wherein the
set of permissions are associated with a user profile.

20 223. The data processing system of claim 215, wherein the
user input is a change in security.

224. A computer program product in a computer readable
medium for managing permissions in an application, the
25 computer program product comprising:

first instructions for receiving a user input
changing a security level for the application at a
container handled by a view controller;

second instructions for generating a view event
30 describing the user input;

third instructions for receiving the view event at
an application mediator;

fourth instructions, responsive to receiving the view event, for sending a request event in response to the receiving the view event; and

5 fifth instructions for receiving at the application mediator, a permission corresponding to the security level, wherein the permission alters an item in the application.

225. A computer program product in a computer readable
10 medium for managing permissions in an application, the computer program product comprising:

first instructions for receiving a user input at a container handled by a view controller, wherein the user input requests a change in permissions in the
15 application;

second instructions for generating a view event describing the user input;

third instructions for receiving the view event at an application mediator;

20 fourth instructions, responsive to receiving the view event, for retrieving by the application mediator, a set of permissions corresponding to the user input;

fifth instructions for sending the set of permissions to the view controller; and

25 sixth instructions for selectively altering content within the container using the set of permissions.

226. A computer program product in a computer readable medium comprising:

30 first instructions for a view controller, wherein the view controller handles presenting a container, handles input to the container, generates a view event in

2025 RELEASE UNDER E.O. 14176

response to a selected user input, and alters content in the container in response to a change in permissions; and second instructions for an application mediator, wherein the application mediator receives the view event, obtains permissions for content in the container in response to the view event indicating a request to change permissions for the content, and sends the permissions to the view controller to set the permissions in the view controller.

345

227. A process in a data processing system for presenting a view to a client, the process comprising the data processing system implemented steps of:

receiving, at an application mediator, a view event from a view controller, wherein the view event describes an action on a displayed container handled by the view controller;

responsive to a requirement that a change in a placement of the displayed container is required, generating a placement event by the application mediator;

determining, by a placement listener, whether the placement event includes an indication that an alternate view is to be generated; and

responsive to a determination that an alternate view is to be generated, sending a call to a method in the view controller to generate the alternate view.

228. The process of claim 227, wherein the alternate view is a hypertext markup language page and wherein a normal view for the container is a component.

229. The process of claim 227, wherein the alternate view

in a data type definition file.

230. The process of claim 227 further comprising:

receiving the call at the view controller;

5 responsive to receiving the call, generating, by the view controller, a markup language version of the displayed container.

231. The process of claim 230 further comprising:

10 sending the markup language version of the displayed container to a client.

232. The process of claim 231, wherein the step of

generating, by the view controller, a markup language
15 version of the displayed container comprises generating a hypertext markup language version of the displayed container.

233. A data processing system comprising:

20 a plurality of view controllers, wherein the plurality of view controllers handle presentation of an interface;

an application mediator, wherein the application mediator handles an order in which the plurality of view
25 controllers are displayed in a display; and

a placement listener, wherein the placement listener handles placement of the interface in the display, wherein the view controller generates a first event handled by the application mediator, the application
30 mediator generates a second event in response to the first event requiring a change by a view controller within the plurality of view controllers, wherein the

Accepted for Publication

placement listener sends a call to the view controller to present the container using an alternate mechanism.

234. The data processing system of claim 233, wherein a
5 view controller is displayed by causing the view controller to present a container in the display.

235. The data processing system of claim 233, wherein the
10 alternate mechanism generates a hypertext markup language file.

236. The data processing system of claim 233, wherein the
15 alternate mechanism is a method in the view controller, wherein the method generates a markup language file representing a display of the interface handled by the view controller.

237. The data processing system of claim 236, wherein the
20 plurality of view controllers, the application mediator, and the placement listener are located on the data processing system and wherein the data processing system is a server.

238. The data processing system of claim 237 further
25 comprising:

sending means for sending the markup language file to a client.

239. A data processing system for presenting a view to a
30 client, the data processing system comprising:

receiving means for receiving, at an application mediator, a view event from a view controller, wherein

SECRET "TOP SECRET"

the view event describes an action on a displayed container handled by the view controller;

generating means, responsive to a requirement that a change in a placement of the displayed container is required, for generating a placement event by the application mediator;

determining means for determining, by a placement listener, whether the placement event includes an indication that an alternate view is to be generated; and

sending means, responsive to a determination that an alternate view is to be generated, for sending a call to a method in the view controller to generate the alternate view.

240. The data processing system of claim 239, wherein the alternate view is a hypertext markup language page and wherein a normal view for the container is a component.

241. The data processing system of claim 239, wherein the alternate view in a data type definition file.

242. The data processing system of claim 239 further comprising:

receiving means for receiving the call at the view controller;

generating means, responsive to receiving the call, for generating, by the view controller, a markup language version of the displayed container.

243. The data processing system of claim 242 further comprising:

sending means for sending the markup language

SECRET "TTBEEH00

version of the displayed container to a client.

244. The data processing system of claim 243, wherein the
5 means of generating, by the view controller, a markup
language version of the displayed container comprises
generating a hypertext markup language version of the
displayed container.

245. A computer program product in a computer readable
10 medium for presenting a view in a data processing system
to a client, the computer program product comprising:

first instructions for receiving, at an application
mediator, a view event from a view controller, wherein
the view event describes an action on a displayed
15 container handled by the view controller;

second instructions, responsive to a requirement
that a change in a placement of the displayed container
is required, for generating a placement event by the
application mediator;

20 third instructions for determining, by a placement
listener, whether the placement event includes an
indication that an alternate view is to be generated; and

fourth instructions, responsive to a determination
that an alternate view is to be generated, for sending a
25 call to a method in the view controller to generate the
alternate view.

246. A computer program product in a computer readable
medium comprising:

30 first instructions for a plurality of view
controllers, wherein the plurality of view controllers
handle presentation of an interface;

second instructions for an application mediator, wherein the application mediator handles an order in which the plurality of view controllers are displayed in a display; and

- 5 third instructions for a placement listener, wherein the placement listener handles placement of the interface in the display, wherein the view controller generates a first event handled by the application mediator, the application mediator generates a second event in response
- 10 to the first event requiring a change by a view controller within the plurality of view controllers, wherein the placement listener sends a call to the view controller to present the container using an alternate mechanism.

3913
247. A method in a data processing system for processing user input in a graphical user interface, the method comprising the data processing system implemented step of:

- 20 presenting a graphical user interface using a view controller, wherein the view controller handles the user input to the graphical user interface;

responsive to a selected user input, sending an event to a first application mediator; and

- 25 responsive to the first application mediator being unable to process the event, sending the event to a second application mediator for processing, wherein the first application mediator and the second application mediator handle an order in which a set of displays are
- 30 displayed by a view controller.

248. The method of claim 247, wherein the first

application mediator provides a first function and the second application mediator provides a second function.

249. The method of claim 247, wherein the selected user
5 input is a selection of a button in the graphical user interface.

250. The method of claim 247, wherein the view controller handles presentation of a container.

10

251. The method of claim 247, wherein the first application mediator is a child of the second application mediator.

15

252. A data processing system comprising:

a plurality of view controllers associated with a set of views, wherein the each view controller within the plurality of view controllers handles a view from the set of views for a graphical user interface and wherein the
20 plurality of view controllers generate events in response to a selected user input to the set of views; and

a plurality of application mediators hierarchically associated, wherein each application mediator handles different functions, a first application mediator on one
25 level within the plurality of application mediators receives an event generated by a view controller, the first application mediator sends the event to a second application mediator on another level within the plurality of application mediators in response to an
30 inability to process the event by the first application mediator.

253. The data processing system of claim 252, wherein the plurality of application mediators include listeners, wherein a first level application mediator includes a listener for events from the plurality of view
5 controllers.

254. The data processing system of claim 253, wherein a second level application mediator includes a listener to the first level application mediator for events from the
10 first level application mediator.

255. The data processing system of claim 252, wherein the first application mediator receives the event from the view controller.
15

256. The data processing system of claim 252, wherein the first application mediator receives the event from another application mediator on another level within the plurality of application mediators.
20

257. The data processing system of claim 252, wherein the plurality of application mediators form a hierarchy from the association, wherein an event first received by an application mediator on a lowest level in the hierarchy
25 and moves up the hierarchy in response to an application mediator in a level of the hierarchy being unable to handle the event.

258. A data processing system for processing user input
30 in a graphical user interface, the data processing system comprising:

presenting means for presenting a graphical user

66207-1130000

interface using a view controller, wherein the view controller handles the user input to the graphical user interface;

first sending means, responsive to a selected user input, for sending an event to a first application mediator; and

second sending means, responsive to the first application mediator being unable to process the event, for sending the event to a second application mediator for processing, wherein the first application mediator and the second application mediator handle an order in which a set of displays are displayed by a view controller.

259. The data processing system of claim 258, wherein the first application mediator provides a first function and the second application mediator provides a second function.

260. The data processing system of claim 258, wherein the selected user input is a selection of a button in the graphical user interface.

261. The data processing system of claim 258, wherein the view controller handles presentation of a container.

262. The data processing system of claim 258, wherein the first application mediator is a child of the second application mediator.

263. A computer program product in a computer readable medium for processing user input in a graphical user

interface, the computer program product comprising:

first instructions for presenting a graphical user interface using a view controller, wherein the view controller handles the user input to the graphical user interface;

second instructions, responsive to a selected user input, for sending an event to a first application mediator; and

third instructions, responsive to the first application mediator being unable to process the event, for sending the event to a second application mediator for processing, wherein the first application mediator and the second application mediator handle an order in which a set of displays are displayed by a view controller.

264. A computer program product in a computer readable medium comprising:

first instructions for a plurality of view controllers associated with a set of views, wherein the each view controller within the plurality of view controllers handles a view from the set of views for a graphical user interface and wherein the plurality of view controllers generate events in response to a selected user input to the set of views; and

second instructions for a plurality of application mediators hierarchically associated, wherein each application mediator handles different functions, a first application mediator on one level within the plurality of application mediators receives an event generated by a view controller, the first application mediator sends the event to a second application mediator on another level

within the plurality of application mediators in response to an inability to process the event by the first application mediator.

- 398
5
265. A data processing system comprising:
a plurality of view controllers, wherein the plurality of view controllers handle display of a plurality of containers and generate events in response to a user input to the plurality of containers; and
- 10 a plurality of application mediators, wherein the plurality of application mediators handle events from the plurality of view controllers, wherein each of the plurality of application mediators include state machine used to manage reception and processing events.
- 15
266. The data processing system of claim 265, wherein a state machine for an application mediator is initiated by reading statements from a data structure.
- 20
267. The data processing system of claim 266, wherein the data structures is a state machine file.
- 25
268. The data processing system of claim 266, wherein the statements define rules for states used to trigger state transitions and actions associated with the state transitions.
- 30
269. The data processing system of claim 268, wherein each of the plurality of application mediators are initiated by reading selected statements from the statements in the data structure.

SECRET

270. A method in a data processing system for presenting a set of screens in a graphical user interface, the method comprising the data processing system implemented steps of:

- 5 presenting a first screen within a set of screens, wherein the set of screens are presented using a set of view controllers;

 responsive to a selected user input to the first screen, generating an event by a view controller within
10 the set of view controllers identifying the user input to the first screen, which is handled by the first view controller; and

 responsive to detecting the event generated by the view controller, selecting, by an application mediator, a
15 second screen from the set of screens for display by sending a response to a view controller handling the second screen.

271. The method of claim 270, wherein the screen is a
20 component.

272. The method of claim 270, wherein the selecting step is performed using a state machine in the application mediator.

25 273. The method of claim 270, wherein the event includes a major code and a minor code.

274. The method of claim 270, wherein the major code indicates an action taken and the minor code indicates a
30 function to be performed.

275. The method of claim 270, wherein the application

SECRET

mediator is initialized by reading a file containing a set of rules.

276. The method of claim 275, wherein the set of rules
5 are a set of transition rules for a state machine.

277. The method of claim 276, wherein the application mediator is initialized using a portion of the set of rules.

10

278. A data processing system comprising:

a plurality of screens presented by a plurality of view controllers, wherein each view controller is associated with a screen, controls presentation of the
15 screen, controls internal operation of the screen, and generates an event in response to a selected input to the screen; and

an application mediator, wherein the application mediator receives events from the plurality of view
20 controllers and provides responses to the plurality of view controllers to alter the display of the plurality of screens.

279. The data processing system of claim 278, wherein the
25 plurality of screens are displayed one screen at a time.

280. The data processing system of claim 278, wherein the plurality of screens are displayed in an order controlled by the application mediator.

30

281. The data processing system of claim 278, wherein the event includes an identification of the user input to a

2025 RELEASE UNDER E.O. 14176

screen and includes data and wherein the application mediator provides a function to process the data.

282. The data processing system of claim 278, wherein the application mediator includes a state machine used to process events.

283. The data processing system of claim 278, wherein the event includes a major code and a minor code.

284. The data processing system of claim 278, wherein the major code indicates an action taken and the minor code indicates a function to be performed.

285. A data processing system for presenting a set of screens in a graphical user interface, the data processing system comprising:

presenting means for presenting a first screen within a set of screens, wherein the set of screens are presented using a set of view controllers;

generating means, responsive to a selected user input to the first screen, for generating an event by a view controller within the set of view controllers identifying the user input to the first screen, which is handled by the first view controller; and

selecting means, responsive to detecting the event generated by the view controller, for selecting, by an application mediator, a second screen from the set of screens for display by sending a response to a view controller handling the second screen.

286. The data processing system of claim 285, wherein the

screen is a component.

287. The data processing system of claim 285, wherein the selecting means is performed using a state machine in the application mediator.

288. The data processing system of claim 285, wherein the event includes a major code and a minor code.

289. The data processing system of claim 285, wherein the major code indicates an action taken and the minor code indicates a function to be performed.

290. The data processing system of claim 285, wherein the application mediator is initialized by reading a file containing a set of rules.

291. The data processing system of claim 290, wherein the set of rules are a set of transition rules for a state machine.

292. The data processing system of claim 291, wherein the application mediator is initialized using a portion of the set of rules.

293. A computer program product in a computer readable medium comprising:

first instructions for a plurality of view controllers, wherein the plurality of view controllers handle display of a plurality of containers and generate events in response to a user input to the plurality of containers; and

second instructions for a plurality of application mediators, wherein the plurality of application mediators handle events from the plurality of view controllers, wherein each of the plurality of application mediators
5 include state machine used to manage reception and processing events.

294. A computer program product in a computer readable medium for presenting a set of screens in a graphical
10 user interface, the computer program product comprising:

first instructions for presenting a first screen within a set of screens, wherein the set of screens are presented using a set of view controllers;

second instructions, responsive to a selected user
15 input to the first screen, for generating an event by a view controller within the set of view controllers identifying the user input to the first screen, which is handled by the first view controller; and

third instructions, responsive to detecting the
20 event generated by the view controller, for selecting, by an application mediator, a second screen from the set of screens for display by sending a response to a view controller handling the second screen.

25 295. A computer program product in a computer readable medium comprising:

first instructions for a plurality of screens presented by a plurality of view controllers, wherein each view controller is associated with a screen,
30 controls presentation of the screen, controls internal operation of the screen, and generates an event in response to a selected input to the screen; and

second instructions for an application mediator, wherein the application mediator receives events from the plurality of view controllers and provides responses to the plurality of view controllers to alter the display of the plurality of screens.

399 296. A method in a data processing system for serializing a data element, the method comprising the data processing system implemented steps of:

- 10 receiving the data element for serialization, wherein data element includes a class name;
replacing the class name with an indicator having a smaller size than the class name to form a modified data element; and
15 serializing the modified data element.

297. The method of claim 296, wherein the step of replacing the class name string with an indicator having a smaller size than the class name to form a modified
20 data element comprises hashing the class name to create a hash code and replacing the class name with the hash code.

298. The method of claim 297 further comprising:
25 receiving the modified data element;
deserializing the modified data element; and
replacing the hash code with the class name string.

299. The method of claim 296, wherein the data element
30 includes a path and wherein the path and the class name string are replaced with the indicator.

300. The method of claim 296 further comprising:
determining whether the data element includes a
class name that is replaceable; and
responsive to a determination that the data is
5 unreplaceable serializing the data element using a
default process.

301. The method of claim 296, wherein the steps receiving
a data element for serialization, wherein data element
10 includes a class name; replacing the class name with an
indicator having a smaller size than the class name to
form a modified data element are performed at the data
processing system; and the steps of receiving the
modified data element; deserializing the modified data
15 element; and replacing the hash code with the class name
are performed at another data processing system.

302. A method in a data processing system for
deserializing a data object, the method comprising:
20 receiving a data element for deserialization;
deserializing the data element; and
replacing an indicator within the data element with
a class name.

25 303. The method of claim 302, wherein the indicator is a
hash code and wherein the step of replacing an indicator
within the data element with a class name comprises:
using the hash code as a key within a hash table to
identify the class name; and
30 replacing the hash code with the class name.

304. The method of claim 303, wherein the hash code also

identifies a base Java class.

305. The method of claim 304, wherein the data is a base Java class path.

5

306. A data processing system comprising:
a serializer having a plurality of modes of operation including:

10

a first mode of operation in which the serializer receives a data element for serialization, wherein the data element includes a class name string;

15

a second mode of operation, responsive to receiving the data element in which the serializer replaces the class name string with a code having a smaller size than the class name string to form a modified data element; and

20

a third mode of operation, responsive to forming the modified data element, in which the serializer serializes modified data element.

25

307. The data processing system of claim 306, wherein the serializer replaces the class name string with a code using a hashing function, wherein the code is a hash code.

30

308. A data processing system for serializing a data element, the data processing system comprising:

receiving means for receiving the data element for serialization, wherein data element includes a class name;

replacing means for replacing the class name with an

indicator having a smaller size than the class name to form a modified data element; and

serializing means for serializing the modified data element.

5

309. The data processing system of claim 308, wherein the means of replacing the class name string with an indicator having a smaller size than the class name to form a modified data element comprises hashing the class name to create a hash code and replacing the class name with the hash code.

10

310. The data processing system of claim 309 further comprising:

15

receiving means for receiving the modified data element;

deserializing means for deserializing the modified data element; and

20

replacing means for replacing the hash code with the class name string.

311. The data processing system of claim 308, wherein the data element includes a path and wherein the path and the class name string are replaced with the indicator.

25

312. The data processing system of claim 308 further comprising:

determining means for determining whether the data element includes a class name that is replaceable; and

30

serializing means, responsive to a determination that the data is unreplaceable, for serializing the data element using a default process.

313. The data processing system of claim 308, wherein the means of receiving a data element for serialization, wherein data element includes a class name; replacing the
5 class name with an indicator having a smaller size than the class name to form a modified data element are performed at the data processing system; and the means of receiving the modified data element; deserializing the modified data element; and replacing the hash code with
10 the class name are performed at another data processing system.

314. A data processing system for deserializing a data object, the data processing system comprising:
15 receiving means for receiving a data element for deserialization;
deserializing means for deserializing the data element; and
replacing means for replacing an indicator within
20 the data element with a class name.

315. The data processing system of claim 314, wherein the indicator is a hash code and wherein the means of replacing an indicator within the data element with a
25 class name comprises:
using means for using the hash code as a key within a hash table to identify the class name; and
replacing means for replacing the hash code with the class name.

30 316. The data processing system of claim 315, wherein the hash code also identifies a base Java class.

317. The data processing system of claim 316, wherein the data is a base Java class path.

- 5 318. A computer program product in a computer readable medium for serializing a data element, the computer program product comprising:

first instructions for receiving the data element for serialization, wherein data element includes a class
10 name;
second instructions for replacing the class name with an indicator having a smaller size than the class name to form a modified data element; and
third instructions for serializing the modified data
15 element.

319. A computer program product in a computer readable medium for deserializing a data object, the computer program product comprising:

- 20 first instructions for receiving a data element for deserialization;
second instructions for deserializing the data element; and
third instructions for replacing an indicator within
25 the data element with a class name.

- 401 // 320. A method in a data processing system for providing an interface to an application for monitoring execution of the application, the method comprising the data
30 processing system implemented steps of:
detecting an event generated by a component;
determining whether the event is an event selected

for monitoring; and

responsive to a determination that the event is an event selected for monitoring, generating a request event, wherein the request event includes data from the event and a destination.

321. The method of claim 320 further comprising:

routing the request event to a destination object through a transporter, wherein the transporter identifies the destination object based on the destination in the request event;

formatting the request event into a form recognized by the destination at the destination object; and

sending the request event to the destination using a destination object.

322. The method of claim 320, wherein the destination is a data structure on a second data processing system.

323. The method of claim 320, wherein the destination is another application used to monitor the application

324. The method of claim 320 further comprising:

performing debugging operations using the data.

325. The method of claim 320, wherein the steps of detecting, determining, generating, routing, formatting, and sending are initiated through a call to an interface in the application.

326. The method of claim 320, wherein the component is a ViewController.

327. The method of claim 320, wherein the component is a Transporter.

5 328. The method of claim 320, wherein the component is a bean.

329. A method in a data processing system for providing an interface to an application for monitoring execution
10 of the application in an object oriented environment, the method comprising the data processing system implemented steps of:

receiving a call to return objects in the application;

15 adding listeners to the objects in the application;
monitoring the objects for an event; and
responsive to detecting an event performing a monitoring operation.

20 330. The method of claim 329, wherein the monitoring operation is a debugging function.

331. The method of claim 329, wherein the monitoring operation is a tracing function.

25 332. An object oriented application in data processing system comprising:

a plurality of view controllers, wherein the plurality of view controllers handle presentation of a
30 plurality of containers in a graphical user interface and generate view events in response to selected user input, wherein a view event includes an identification of an

Docket No. A19-99-339

action taken on a container;

an application mediator, wherein the application mediator includes a listener for receiving view events from the plurality of view controllers, processes the view events, and selectively generates request events based on the view events, wherein each request event includes a first identifier identifying a destination and a second identifier identifying a function to be performed;

10 a plurality of destination objects, wherein each destination object is associated with a destination and is configured to translate a request event into a form recognized by the destination to form a translated request event and send the translated request event to the destination;

a transporter, wherein the transporter receives view events from the application mediator and routes the view events to a destination within the plurality of destinations based on the first identifier; and

20 a snoop application mediator, wherein the snoop application mediator, includes a listener for receiving view events from the plurality of view controllers and generates request events having a data structure used to monitor the object oriented application as a destination, wherein the data being monitored is sent to the data structure.

333. The object oriented application of claim 332, wherein the snoop application mediator includes a listener for receiving request events from the application mediator.

335. The object oriented application of claim 333,
wherein the data includes select view events and the data
structure is a log file.

337. The object oriented application of claim 333,
wherein the data is an exception generated by the
15 application mediator.

20 detecting means for detecting an event generated by
a component;

339. The data processing system of claim 338 further
30 comprising:

routing means for routing the request event to a destination object through a transporter, wherein the

Docket No. A19-99-339

transporter identifies the destination object based on the destination in the request event;

formatting means for formatting the request event into a form recognized by the destination at the

5 destination object; and

sending means for sending the request event to the destination using a destination object.

340. The data processing system of claim 338, wherein the
10 destination is a data structure on a second data processing system.

341. The data processing system of claim 338, wherein the
15 destination is another application used to monitor the application

342. The data processing system of claim 338 further comprising:

performing means for performing debugging operations
20 using the data.

343. The data processing system of claim 338, wherein the detecting means, determining means, generating means, routing means, formatting means, and sending means are
25 initiated through a call to an interface in the application.

344. The data processing system of claim 338, wherein the component is a ViewController.
30

345. The data processing system of claim 338, wherein the component is a Transporter.

346. The data processing system of claim 338, wherein the component is a bean.

5 347. A data processing system for providing an interface to an application for monitoring execution of the application in an object oriented environment, the data processing system comprising:

receiving means for receiving a call to return
10 objects in the application;
adding means for adding listeners to the objects in the application;
monitoring means for monitoring the objects for an event; and
15 performing means, responsive to detecting an event, for performing a monitoring operation.

348. The data processing system of claim 347, wherein the monitoring operation is a debugging function.

20 349. The data processing system of claim 347, wherein the monitoring operation is a tracing function.

350. A computer program product in a computer readable
25 medium for providing an interface to an application for monitoring execution of the application, the computer program product comprising:

first instructions for detecting an event generated by a component;
30 second instructions for determining whether the event is an event selected for monitoring; and
third instructions, responsive to a determination

that the event is an event selected for monitoring, for generating a request event, wherein the request event includes data from the event and a destination.

5 351. A computer program product in a computer readable medium for providing an interface to an application for monitoring execution of the application in an object oriented environment, the computer program product comprising:

10 first instructions for receiving a call to return objects in the application;

second instructions for adding listeners to the objects in the application;

15 third instructions for monitoring the objects for an event; and

fourth instructions responsive to detecting an event for performing a monitoring operation.

20 352. A computer program product in a computer readable medium:

25 first instructions for a plurality of view controllers, wherein the plurality of view controllers handle presentation of a plurality of containers in a graphical user interface and generate view events in response to selected user input, wherein a view event includes an identification of an action taken on a container;

30 second instructions for an application mediator, wherein the application mediator includes a listener for receiving view events from the plurality of view controllers, processes the view events, and selectively generates request events based on the view events,

2025 RELEASE UNDER E.O. 14176

wherein each request event includes a first identifier identifying a destination and a second identifier identifying a function to be performed;

third instructions for a plurality of destination
5 objects, wherein each destination object is associated with a destination and is configured to translate a request event into a form recognized by the destination to form a translated request event and send the translated request event to the destination;

10 fourth instructions for a transporter, wherein the transporter receives view events from the application mediator and routes the view events to a destination within the plurality of destinations based on the first identifier; and

15 fifth instructions for a snoop application mediator, wherein the snoop application mediator, includes a listener for receiving view events from the plurality of view controllers and generates request events having a data structure used to monitor the object
20 oriented application as a destination, wherein the data being monitored is sent to the data structure.

402
353. A process in a data processing system for processing
an event, the method comprising the data processing
25 system implemented steps of:

generating the event at a object in an object oriented environment, wherein the event includes a class name as a destination and a method name as a function to be invoked; and

30 sending an event from the object to a second object oriented environment, wherein the event is used to access a method in the second object oriented environment.

354. The process of claim 353, wherein the event further includes parameters for the method as data.

5 355. The process of claim 353, wherein the access is an remote invocation of the method.

356. The process of claim 353, wherein the object is an application mediator.

10

357. The process of claim 356, wherein the step of sending the event to the second environment comprises:

15 sending the event to a destination object, wherein the destination object translates the event into a format recognized in the second object oriented environment.

358. A process in a data processing system for processing events an object oriented system, the method comprising the data processing system implemented steps of:

20 responsive to receiving a selected user input to a container, sending a view event from a view controller to an application mediator, wherein the view event identifies an action taken to generate the selected user input;

25 selectively generating a request event based on the view event, wherein the request event includes a major code identifying a class name as a destination and a minor code identifying a method name a function to be invoked; and

30 sending the request event to a transporter; and responsive to receiving the request event at the transporter, sending the request event to a destination

object within a plurality of destination objects based in the class name.

359. The process of claim 358, wherein the action is a pressing of a button displayed in the container.

360. The process of claim 358, wherein each of the plurality of destination objects is associated with a destination and wherein each of the plurality of destination objects translates the request event into a format recognizable by the destination.

361. The process of claim 360, wherein the plurality of destinations is a plurality of applications located on a remote data processing system having an ability to communicate with the data processing system.

362. The process of claim 361, wherein the request event is used to invoke the method at the destination.

363. The process of claim 358, wherein the request event includes data in a form of parameters for the method.

364. The process of claim 361, wherein the request event is used to alter a method at the destination

365. The process for claim 361, wherein the request is event is used access a class located at the destination.

366. A data processing system for processing an event, the data processing system comprising:
generating means for generating the event at a

object in an object oriented environment, wherein the event includes a class name as a destination and a method name as a function to be invoked; and

5 sending means for sending an event from the object to a second object oriented environment, wherein the event is used to access a method in the second object oriented environment.

367. The data processing system of claim 366, wherein the event further includes parameters for the data processing system as data.

368. The data processing system of claim 366, wherein the access is an remote invocation of the data processing system.

369. The data processing system of claim 366, wherein the object is an application mediator.

20 370. The data processing system of claim 369, wherein the means of sending the event to the second environment comprises:

25 sending means for sending the event to a destination object, wherein the destination object translates the event into a format recognized in the second object oriented environment.

371. A data processing system for processing events in an object oriented system, the data processing system comprising:

30 first sending means, responsive to receiving a selected user input to a container, for sending a view

event from a view controller to an application mediator, wherein the view event identifies an action taken to generate the selected user input;

generating means for selectively generating a
5 request event based on the view event, wherein the request event includes a major code identifying a class name as a destination and a minor code identifying a method name a function to be invoked; and

second sending means for sending the request event
10 to a transporter; and

third sending means, responsive to receiving the request event at the transporter, for sending the request event to a destination object within a plurality of destination objects based in the class name.

15

372. The data processing system of claim 371, wherein the action is a pressing of a button displayed in the container.

20

373. The data processing system of claim 371, wherein each of the plurality of destination objects is associated with a destination and wherein each of the plurality of destination objects translates the request event into a format recognizable by the destination.

25

374. The data processing system of claim 373, wherein the plurality of destinations is a plurality of applications located on a remote data processing system having an ability to communicate with the data processing system.

30

375. The data processing system of claim 374, wherein the request event is used to invoke the data processing

66601-113046

system at the destination.

376. The data processing system of claim 371, wherein the request event includes data in a form of parameters for the data processing system.

377. The data processing system of claim 374, wherein the request event is used to alter a data processing system at the destination

378. The data processing system for claim 374, wherein the request is event is used access a class located at the destination.

379. A computer program product in a computer readable medium for processing an event, the computer program product comprising:

first instructions for generating the event at a object in an object oriented environment, wherein the event includes a class name as a destination and a method name as a function to be invoked; and

second instructions for sending an event from the object to a second object oriented environment, wherein the event is used to access a method in the second object oriented environment.

380. A computer program product in a computer readable medium for processing events an object oriented system, the computer program product comprising:

first instructions, responsive to receiving a selected user input to a container, for sending a view event from a view controller to an application mediator,

Docket No. 99-339

wherein the view event identifies an action taken to generate the selected user input;

second instructions for selectively generating a request event based on the view event, wherein the

- 5 request event includes a major code identifying a class name as a destination and a minor code identifying a method name a function to be invoked; and

third instructions for sending the request event to a transporter; and

- 10 fourth instructions, responsive to receiving the request event at the transporter, for sending the request event to a destination object within a plurality of destination objects based in the class name.

15

2000-11-16 14:00:00